链滴

`var`、`let`、`const` 和没有关键字的声明有什么区别?

作者：Vanessa

原文链接：https://ld246.com/article/1549635108016

来源网站：链滴

2019-02-08

回答

无关键字

在变量赋值之前如果没有关键字的话，则会把变量分配给全局变量或覆盖已经声明的变量。在非严格式下，如果变量还没有被声明的话，他将会把变量做为全局对象（浏览器中的 window）的一个属性在严格模式下，他将抛出异常以防止创建不需要的全局变量。

var

var 是 ES2015 以前声明变量的默认语句。他在函数作用域内创建的变量可以在该作用域中被重新赋和重新声明。但是，由于缺少块作用域，变量在块作用域外将继续存在。如果在含有循环的异步回调变量被重用的话将会产生问题。

以下代码片断中，在执行 setTimeout 回调时，循环完已经完成且变量 i 变为了 10，因此十个回调都用了函数作用域中的同一个变量。

```
for (var i = 0; i < 10; i++) {
  setTimeout(() => {
    console.log(i) // 10
  })
}
```

可以通过消除块作用域或创建一个新的函数作用域来解决此问题

```
for (var i = 0; i < 10; i++) {
  setTimeout(console.log, 1000, i)
}

for (var i = 0; i < 10; i++) {
  ;(i => {
    setTimeout(() => {
      console.log(i)
    })
  })(i)
}
```

let

let 是在 ES2015 中引入的，他是一种可在变量声明后可再赋值的常用声明方式。再次声明相同的变将会抛出异常。他是有块作用域的，因此在循环中使用时将会保持在同一个作用域下迭代。

```
for (let i = 0; i < 10; i++) {
  setTimeout(() => {
    console.log(i) // 0 1 2...9
  })
}
```

const

const 是在 ES2015 中引入的，他是一种新的默认的常用的声明方式。他声明的所有变量将不可再被新赋值，如果是对象的话，必须保持对象的引用不变。他是块作用域的，且不能被再次赋值。

```
const myObject = {}
myObject.prop = "hello!"
myObject = "hello" // Uncaught TypeError: Assignment to constant variable.
```

加分回答

- 所有声明在其范围内都会被提升。
- let 和 const 中有一个称为时间死区（temporal dead zone TDZ）的概念。虽然声明会被提升，在进入作用域之后、声明之前他将无法被访问。
- 尽可能避免使用 var，将 const 作为所有变量的默认声明语句，如果后面需要对变量进行重新分就使用 let。

返回总目录

每天 30 秒